Iowa Department of Public Health
Promoting and Protecting the Health of Iowans

Electronic Lab Reporting

IDPH Simplified ELR Message Format Specification

Messaging protocol for electronic reporting of communicable diseases to the Iowa Department of Public Health

Version 2.0

Revision History

Date	Version	Description	Author
03/27/2013	1.0	Published	Ahmad Kasmieh, John Satre, Swathi Ramasahayam
08/10/2021	2.0	Published	Ahmad Kasmieh, John Satre, Swathi Ramasahayam
ORC		Added details for: 1. Ordering provider NPI (ORC 12.1) 2. Order Effective Date (ORC 15)	John Satre, Swathi Ramasahayam
OBR		Added Test type (OBR 4.1)	John Satre, Swathi Ramasahayam
OBX		Added Device ID (OBX 17.1)	John Satre, Swathi Ramasahayam

©Iowa Department of Public Health, 2021

Table of Contents

 ©Iowa Department of Public Health, 2021

Introduction

This document describes the Simplified Message Format (SMF) standard published by the Iowa Department of Public Health (IDPH) for the electronic submission of reportable conditions. The SMF was created for use by hospitals with limited IT resources as a simplified alternative to the Meaningful Use HL7 2.5.1 ORU (HL7) standard. Ultimately, the Department of Public Health would like to see every hospital submit messages using the HL7 standard protocol. The following table provides a brief guide to choosing which messaging standard is right for your hospital.

	Encoding Style	**Implementation Approach**	**Message Complexity**
SMF	Pipe delimited fields	Simple string concatenation	Low
HL7	XML	Custom code with XML API or Map to HL7 Interface tool	High if coding by hand Moderate to low if using HL7 interface software
HL7	EDI	Complex string concatenation or Map to HL7 Interface tool	High if coding by hand Moderate to low if using HL7 interface software

■ Table 1 - Supported Message Format Overview

While this document is primarily an implementer's guide for the SMF, it is also useful to HL7 implementers given that it highlights which HL7 fields are required by IDPH and provides an HL7 implementer's addendum that covers the HL7 batch protocol. HL7 implementers should refer to the IDPH publication Iowa Department of Public Health Implementation Guide and ELR

Constrained profile found at the URL http://www.idph.state.ia.us/adper/common/pdf/idss/elr_constrained_profile.pdf

Message Structure

A message consists of a series of discrete segments. Each segment contains distinct information related to the infectious disease report and segments follow a defined order. The SMF's six segments are adapted from segments defined in the HL7 ORU message. Each segment contains a fixed set of columns. If a segment is defined as having seven columns, it must be sent with delimiters for all seven columns regardless of the presence of data within the column. The data within the columns can be of variable length from zero characters to the max length specified for a column in the message specification. Each segment is prefixed with its segment identifier string and terminated with a carriage return line feed combination (Hex /D0/A0). Columns within a segment are delimited using the pipe character (|). A collection of properly ordered segments constitutes a message.

A typical Batch Header Segment (BHS) segment looks as follows:

```
BHS|1.0| IA Public Health Lab|12D3456789|20050614|2005061422D0650270↵
```

- Table 2 - Sample BHS Segment

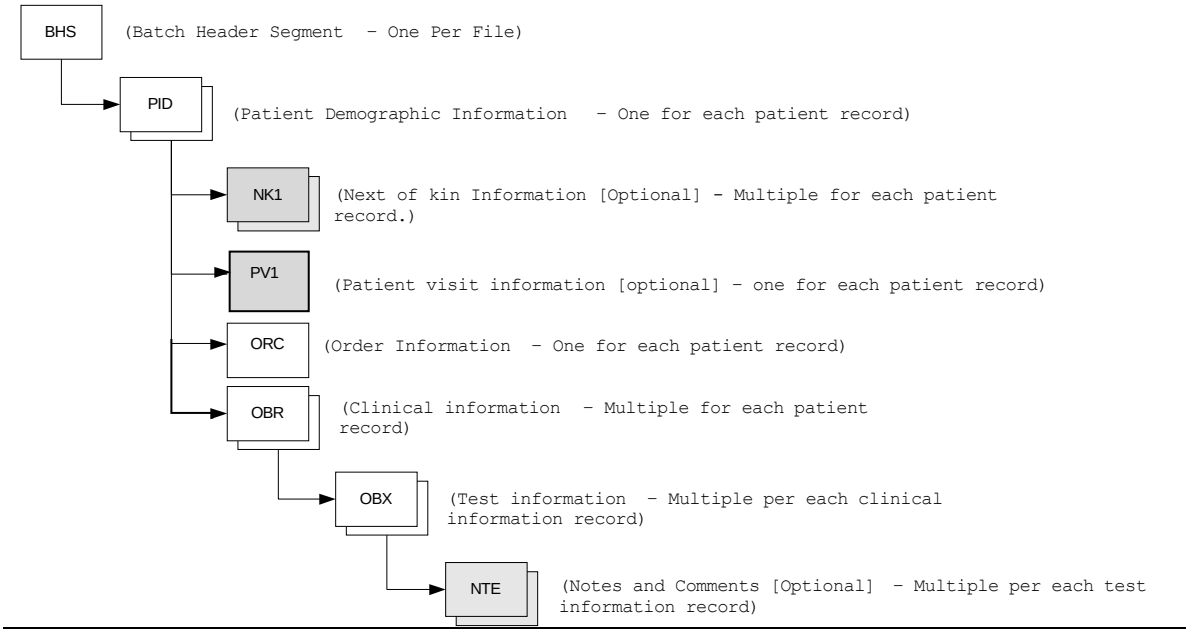
Note that the terminal delimiter of the BHS segment is a carriage return and not a pipe. Pipe delimiters should only be used between columns in the message.

The message specification defines seven segment types identified by the following prefixes: BHS, PID, NK1, ORC, OBR, OBX and NTE. The segments must appear in a specific sequence within the message in order for it to be successfully processed. A message file will contain one header segment (BHS) and multiple reportable event segment collections. Each reportable event is described using a combination of the PID (Patient Demographic Information), NK1 (Next of Kin), PV1 (Patient Visit Information), ORC (Order Information), OBR (Clinical Information), OBX (Test Information) and NTE (Notes and Comments) segments.

Figure 1 depicts the hierarchical relationship between the segments through the use of a tree diagram. Each segment is represented using a box and relationships between segments are represented using an arrow. Segments are depicted in order from top to bottom so that, for example, OBR always follows ORC and OBX always follows OBR. If a segment can repeat, it is represented using stacked boxes. If a segment is optional, it is represented using a box with a gray fill.

Using parent/child terminology it can be said that a parent PID segment can have child NK1, PV1, ORC and OBR segments that are exclusively tied to it and contain information relevant only to the parent PID. The child segments are related to their parent PID solely by their position in the file.

 ©Iowa Department of Public Health, 2021



■ Figure 1 - Hierarchy of Segments in Message Batch File

NOTE: The Arabic numeral "1" in the segment name NK1 and PV1 does not denote a numbering scheme. The 1 is literally part of the segment's name. If a message contained three NK1 segments, they would all be prefixed with NK1 and if the message had three PV1 segments, they would all be prefixed with PV1

Figure 2 depicts an example batch file. A typical batch file might contain a segment ordering similar to what is seen in Figure 2. *Note that for brevity, the columns that would follow each segment prefix have been omitted.* Additionally, the numbers in italic would not appear in the file. They are only given as an aid to the discussion of the file structure.

Line Position In File	**Segment Name**
1	BHS
2	**PID**
3	NK1
4	NK1
5	PV1
6	ORC
7	OBR
8	OBX
9	OBR
10	OBX
11	OBX
12	**PID**
13	NK1
14	PV1
15	ORC

16	OBR
17	OBX
18	NTE

■ Figure 2 - Example Valid Segment Sequence

In this file, the series of segments describes two patient records. The boundaries of these records are identified by the prefixes on lines 2 and 12 of the file. That is to say that all segments on lines 2 through 11 inclusive contain clinical data for the patient identified by the PID segment at line 2 and segments 12 through 18 inclusive contain clinical data for the patient identified by the PID at line 12.

It follows that for each parent-child relationship identified by the hierarchy description in Figure 2 these exact rules are followed. Therefore the OBR segment on line 9 is the parent to the OBX children on lines 10 and 11. The end of the relationship is noted by the occurrence of either a new OBR segment or a PID segment signaling the beginning of a new patient's record.

Note: A valid sequence for a batch upload might consist of the BHS segment with no other segments in cases where no instances of reportable results were found.

Segment Column Definitions

The following section presents a collection of tables defining the acceptable set of columns for each of the segments defined in the Simplified ELR Message. The column order specified in the tables must be followed exactly in the message sent to IDPH. Additionally, columns are identified as Required, Conditional or Requested. These categories refer to data and not delimiters. For instance, take a hypothetical segment HYP consisting of three columns the first and third of which are required and the second of which is optional. If the reporting institution creating the message does not have a value in its system to populate the second column then the HYP segment must be constructed as follows:

HYP|First||Third↵

Note that the delimiters for the second column are present but the second column itself it empty. The rules for Required, Conditional, Requested and Optional columns are broken down in Table 4.

Key to Column Descriptions

Use Table 3 as a guide to the meaning of the columns in each of the tables that describe the message segments.

Column Name	**Purpose**
Column Name	Describes the data to be included in the segment for the purposes of electronic laboratory reporting.
Length	The maximum length of data in the column.
Data Format	Identifies the data type for the column. Refer to Table 8 for definitions of each data type.
Required?	Refer to Table 4

 ©Iowa Department of Public Health, 2021

Column Name	Purpose
Mapped	Flags columns that must have local-code to universal code mappings defined via using IDPH provided web-mapping interface. It is expected that these columns will be populated with local codes when transmitted.
HL7 Component	Refers to the segment component in the HL7 2.3.1 standard ORU message type. Provided to assist HL7 implementers. SMF implementers can disregard this column.

- Table 3 - Column Descriptions

Table 4 lists the possible values for the "Required?" column and provides a definition for each value.

Category	Definition
Required	Value must be present in order for the message to successfully validate and process within the IDPH ELR System.
Conditional	Value must be present in certain instances, see tables that follow OBR and OBX segment column definitions.
Requested	Value is requested if data is available from the transmitting application; blanks will not affect message transmission or validation.

- Table 4 - Required, Conditional and Requested Column Rules

Understanding Coding Systems

Many of the requested data elements are coded fields that contain values that come from a defined set of accepted values. A code system is a formalized method for describing things in a way that is unambiguous and standardized. The three primary code systems used in ELR are LOINC (for test types), SNOMED (for results) and HL7 (for vocabularies).

Most hospitals use internally developed coding systems. While useful within the hospital's IT system, the local codes are often not interpretable to outside organizations. IDPH requires that local codes be mapped to standard coding systems using either the IDPH Mapping interface or via means available within the hospital's IT environment.

Understanding Institution and Clinician Level Data

Each infectious disease report may contain references to many institutions and clinicians who are involved with the patient, the laboratory testing and the message transmission. Table 5 lists these entities and defines each of them.

Entity	Definition
Sending Facility	The reporting institution where the electronic laboratory report to IDPH originates.
Facility1	In instances where a patient is transferred between facilities, up to two facilities and medical record numbers may be included using these two fields.
Facility2	
Primary Provider	The patient's primary doctor for routine medical care.
Treating Physician	The physician treating the patient at the time of testing.
Ordering Facility	The medical office or hospital where the laboratory test order originated
Ordering Provider/Physician	The physician with whom the laboratory test order originates.
Producing Facility	The facility that houses the laboratory where the test result originates.

- Table 5 – Definitions for entities recognized in ELR messages.

BHS Segment

Provide one BHS per each file transmission of 0 to n patient records.

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
SMF Version	10	ST	Required		N/A
Sending Facility Name	20	ST	Required		4.1
Sending Facility ID (CLIA)	199	ST	Required		4.2
Message Date/Time	26	TS	Required		7
Message control ID (Date/Time + Institution ID)	199	ST	Required		10

Note: Message submitters following this guide will place the value "1.0" in their SMF Version column.

PID Segment

Provide one PID per each patient record.

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Patient Identifier[1]	15	ST	Required		3.1
Assigning Authority ID(CLIA)[2]	199	ST	Required		3.4.2
~~Medical Record Number1~~	~~20~~	~~ID~~	~~Requested~~		~~3.1~~
~~Facility1~~	~~100~~	~~ST~~	~~Requested~~		~~3.6~~
~~Medical Record Number2~~	~~20~~	~~ID~~	~~Requested~~		~~3.1~~
~~Facility2~~	~~100~~	~~ST~~	~~Requested~~		~~3.6~~
~~Social Security Number – Last four digits only~~	~~16~~	~~ST~~	~~Requested~~		~~3.1~~
~~Visit ID~~	~~199~~	~~ST~~	~~Requested~~		~~3.1~~
~~Patient Account~~	~~199~~	~~ST~~	~~Requested~~		~~3.1~~
Record Control Id (Batch Message Control ID + PID sequence number)	100	ST	Required		N/A
Last Name	50	ST	Required		5.1.1
First Name	30	ST	Required		5.2
Middle Initial	30	ST	Requested		5.3
Suffix	20	ST	Requested		5.4
Date of Birth[3]	26	TS	Requested		7
Sex	30	ID	Requested	Y	8
Race	30	ID	Requested	Y	10
Address	120	ST	Requested		11.1.1
Address other Designation	120	ST	Requested		11.2
City	50	ST	Requested		11.3
State	50	ST	Requested		11.4

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Zip	12	ST	Requested		11.5
Country	3	ST	Requested		11.6
Phone Number	20	XTN	Requested		13
Marital Status[4]	30	ID	Requested		16
Ethnicity	30	ID	Requested	Y	22
Death Date	26	TS	Requested		29
Death Indicator	1	ID	Requested	Y	30
Pregnancy Status	30	ID	Requested[2]	Y	OBX 5

PID Segment Column Clarifications

1 – Patient Identifier is a required field. It is used to uniquely identify the patient. This can be a Medical Record Number or a Patient Internal Identifier and it should be generated only once per patient and is always unique to the patient.2 – Assigning Authority ID is a required field. The Assigning Authority is used to identify the system, application or organization that assigned the ID Number (Patient Identifier)

3– Date of Birth is a required field but you may encounter situations when the patient's birth date information is not available. One instance is the case where only patient age is known. Under such circumstances, it is acceptable to send the four-digit year identifier produced by subtracting the patient's age from the current year and January 1 of that year as the estimated date of birth.

4– Marital Status is requested by Iowa, as this element is legally required in Iowa – Need to send it if you have it,

5– Pregnancy status is legally required in Iowa for female patients and requested especially when reporting one of the following organisms:

Chlamydia, COVID-19, Diphtheria, Gonorrhea, Haemophilus influenzae type B invasive disease, Hepatitis B, Hepatitis C, Listeria, Measles, Mumps, Pertussis, Polio, Rubella, Streptococcus pneumonia invasive disease, Syphilis, Tetanus.

NK1 Segment

Repeat 0 to n times for next of kin, primary provider, treating physician. NK1 is an optional segment.

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Last Name	50	ST	Required		2.1.1
First Name	30	ST	Required		2.2
Middle Initial	30	ST	Requested		2.3
Suffix	20	ST	Requested		2.4
Relationship	30	ID	Requested	Y	3
Address	120	ST	Requested		4.1.1
Address other Designation	120	ST	Requested		4.2

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
City	50	ST	Requested		4.3
State	50	ST	Requested		4.4
Zip	12	ST	Requested		4.5
Country	3	ST	Requested		4.6
Phone Number	20	XTN	Requested		5

PV1 Segment

Provide 0 or 1 PV1 segment per each patient record. PV1 is an optional segment,

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Patient Class	20	ID	Required		2
Admit Date/Time	26	TS	Requested		44
Discharge Date/Time	26	TS	Requested		45

ORC Segment

Provide one ORC per each patient record.

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Ordering Provider NPI	10	ST	Requested		12.1
Order Effective Date/Time	26	TS	Requested		15
Ordering Facility Name	100	ST	Required		21.1
Ordering Facility Address	120	ST	Required		22.1.1
Ordering Facility Other Designation	120	ST	Requested		22.2
Ordering Facility City	50	ST	Requested		22.3
Ordering Facility State	50	ST	Requested		22.4
Ordering Facility Zip	12	ST	Requested		22.5
Ordering Facility Country	3	ST	Requested		22.6
Ordering Facility Phone Number	20	XTN	Required		23.5
Ordering Provider Last Name	50	ST	Requested		12.2.1
Ordering Provider First Name	30	ST	Requested		12.3
Ordering Provider Middle Initial	30	ST	Requested		12.4
Ordering Provider Suffix	20	ST	Requested		12.5
Ordering Provider Phone Number	20	XTN	Requested		14
Ordering Provider Address	120	ST	Requested		24.1.1
Ordering Provider Other Designation	120	ST	Requested		24.2
Ordering Provider City	50	ST	Requested		24.3
Ordering Provider State	50	ST	Requested		24.4
Ordering Provider Zip	12	ST	Requested		24.5
Ordering Provider Country	3	ST	Requested		24.6

 ©Iowa Department of Public Health, 2021

OBR Segment

Provide one OBR per each test or battery ordered per patient record.

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Filler Order Number (Laboratory Accession Number)	199	ST	Required		3.1
Test Type Requested Standard LOINC Code	20	ST	Required		4.1
Specimen Collection Date	26	TS	Required		7
Onset Date	26	TS	Requested		OBX 5
Symptoms	300	ST	Requested		13
Specimen receipt date	26	TS	Required		14
Specimen Source	199	ID	Required	Y	15
Ordering Provider NPI	10	ST	Requested		16.1
Report Status Code	1	ID	Required	Y	25
Parent Test Type	20	ID	Optional	Y	26.1
Parent Result	20	ST	Conditional[1]	Y	26.3
Parent Specimen Number (Filler Order Number from Parent)	199	ST	Conditional[1]		29

OBR Column Clarifications

Column Name	Clarification
Placer Order Number	For use in cases where the order number differs from the laboratory accession number
Filler Order Number (Laboratory Accession Number)	Laboratory Accession Number
Test Type Requested Standard LOINC Code	Indicates the test or battery requested by the physician
Report Status Code	The status of the results for this order (preliminary, final, corrected)
Parent Test Type	Provides links to previously performed tests related to this order. Parent test, result and laboratory accession number of previously performed tests related to this specimen. See the section entitled Creating Associations Between Test Results for more details.
Parent Result	
Parent Specimen Number (Filler Order Number from Parent)	

OBR Conditional Data Notes

	Instance	Data Required
1	Message transmitted is an additional test performed on a single specimen.	Parent Result (Initial Result), Parent Specimen Number (Initial Specimen Number)

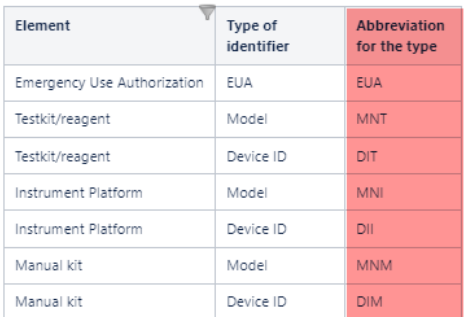
OBX Segment

Provide one to n OBX segments per each OBR per patient record.

 ©Iowa Department of Public Health, 2021

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Test Type Performed Local Code	20	ST	Required	Y	3.4
Result Local Code	20	ST	Conditional[2]	Y	5.4
Result Numeric Value	30	SN	Requested		5.2
Units	20	ID	Conditional[3]		6
Reference Range	60	ST	Conditional[3]		7
Abnormal Flags	20	ID	Requested	Y	8
Observation Result Status	1	ID	Required	Y	11
Test/Result Date	26	TS	Required		14
Producing Facility (CLIA)	60	ST	Required		15.1
Device ID	200	ST	Requested		17.1

OBX Column Clarifications

Column Name	Clarification
Test Type Performed Local Code	Code for the test performed
Result Character Value	Code for the result of the test performed
Result Numeric Value	Numerical Result of the test performed
Units	Units for the result value
Reference Range	Reference range for interpretation of the result value
Abnormal Flags	Interpretations of the result value
Observation Result Status	Status of laboratory result (Preliminary, Final, Corrected)
Device ID	formula to construct the string based identifier: <Model name or Device ID>_<Manufacturer name>_<Abbreviation for the type> Ex: ID Now_Abbott_DII Abbreviation type is drawn from the table below: 

Element	Type of identifier	Abbreviation for the type
Emergency Use Authorization	EUA	EUA
Testkit/reagent	Model	MNT
Testkit/reagent	Device ID	DIT
Instrument Platform	Model	MNI
Instrument Platform	Device ID	DII
Manual kit	Model	MNM
Manual kit	Device ID	DIM

OBX Conditional Data Notes

	Instance	Data Required
2	Test results transmitted are coded values	Local Code Value for Result (Mapped to SNOMED)
3	Numeric Test results are transmitted	Result value, Units, Reference Range

■ Table 6 - Conditional Data Rules

NTE Segment

Provide 0 to n NTE segments per OBR or OBX segment.

 ©Iowa Department of Public Health, 2021

Column Name	Length	Data Format	Required?	Mapped	HL7 Component
Comment	64k	FT	Optional		3

©Iowa Department of Public Health, 2021

Creating Associations Between Test Results

IDPH is considering each OBR as its own test and NOT using the parent-child relationship.However, this message specification allows for the creation of relationships between test results through the use of the "parent" result information columns defined in the OBR segment. As an example, consider a test resulted as a positive Staphylococcus infection. Antibiotic sensitivity tests are ordered as a follow up to the confirmation of the infection. When the antibiotic sensitivity test results are reported, they reference the test result that initially provided confirmation of the infection.

Information that uniquely identifies the confirmatory Staphylococcus test result is taken from the OBR and OBX pair and echoed in the sensitivity test battery's OBR segment in order to establish a relationship between the two results. Table 7 provides a map between the columns in the parent result message and the child result message.

Parent Columns		**Sensitivity or Liver Function Columns**
OBR Filler Order Number	➡	OBR Parent Specimen Number
OBX Test Type Performed Local Code	➡	OBR Parent Test Type
OBX Result	➡	OBR Parent Result

■ Table 7 – OBR Parent Field Mappings

Linking tests applies to sensitivity test results and liver function test results. The next two sections discuss the particulars of each type of test.

Sensitivity Test Results

When sensitivity results are submitted they must be linked to a previously sent condition test result via the use of the OBR segment's Parent Test Type, Parent Result and Parent Specimen Number columns. Table 7 provides a mapping from the Parent Test Result message segment columns to the Sensitivity Result segment columns.

Liver Function Test Results

Report aspartate amino transferase and alanine transaminase quantitative results with reference ranges in conjunction with a positive hepatitis A, B, C, D or E or a positive TB result. Liver function tests related to a hepatitis result should include any testing within the past month. Liver function tests related to a tuberculosis result should include any testing done on the same day as the TB test.

When liver function results are submitted they must be linked to a previously sent condition test result via the use of the OBR segment's Parent Test Type, Parent Result and Parent Specimen Number columns. Table 7 provides a mapping from the Parent Test Result message segment columns to the Liver Function segment columns.

Data Format Guide

 ©Iowa Department of Public Health, 2021

Data Type	Description	Length	Notes	HL7 Reference Number
ID	Coded Value for HL7 defined tables	199	Follows the formatting rules for an ST field except that it is drawn from a table of legal values.	2.8.21
FT	Formatted Text Data	64k	Arbitrary length	2.8.19
ST	String	199	Any printable ASCII characters except the defined delimiter characters; left justified	2.8.40
TS	Time Stamp	18	YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]]]	2.8.44
XTN	Extended Telecommunication Number	19	[(999)]999-9999[X99999]	2.8.52

- Table 8 - Data Type Definitions

NOTE: The square brackets seen in the Notes column of Table 8 denote optional message parts. They are not part of the required message formatting. Using the phone number mask [(999)]999-9999[X99999] tells us that all the following numbers are valid: 231-5555 and (212)231-5555 and 231-5555X234.

 ©Iowa Department of Public Health, 2021

Addendum: HL7 Message Submitter's Guide

Introduction

This addendum provides information specific to submitting observations using the HL7 2.3.1 ORU message. The previous section serves as a guide to the HL7 2.3.1 elements expected by IDPH in observation messages. The IDPH Simplified Message Format Specification is based on the HL7 2.3.1 ELR Implementation Guide from the CDC. All the information necessary to link the requested information to the appropriate ORU message segments and segment components remains in the message tables presented above.

Message Structure

Files submitted to IDPH must follow the HL7 Batch Protocol. The batch protocol specifies a number of header segments used to organize a batch file for processing. Table 9 describes the expected batch protocol segments and their purposes.

The batch protocol recognizes three levels of organization. The top level is the file (File Header Segment), which is assumed to originate from one entity and therefore occurs only once within a file. The next level is the batch (Batch Header Segment). The batch header allows for the fact that a consortium of reporting institutions may send their information in the same file. This information must be segregated in some way and the batch header serves this purpose by bounding the messages of each reporting institution represented in the file transmission. The last level is the message level (Message Header Segment). The message bounds an individual HL7 message.

Segment Name	Rule
FHS (File)/FTS	Included Once Per Message. Use to identify the creator and sender of the file.
BHS (Batch)/BTS	Included for every individual reporting institution represented in the file.
MSH (Message)	Included for every message within the file. MSH is defined as part of a message and therefore does not serve to bound the message. There is no message tail segment.

- Table 9 - HL7 Batch Protocol Header and Tail Segments

Trailer segments are used for bounding the regions defined by a File Header Segment and the Batch Header Segments within a file. These segments serve the additional purpose of providing an area to store auditing information such as the number of batches within a file or the number of messages within a batch. Such information is useful on the processing side to verify that all messages expected were in fact received and processed.

 ©Iowa Department of Public Health, 2021

Header Segment Summary

FHS Segment

SEQ	LEN	DT	R/O	ITEM#	ELEMENT NAME	ELR Usage
1	1	ST	R	67	File field separator	Supported "|"
2	5	ST	R	68	File encoding characters	Supported "^~\&"
3	20	ST	O	69	File sending application	Supported
4	20	ST	O	70	File sending facility	Supported
5	20	ST	O	71	File receiving application	Supported
6	20	ST	O	72	File receiving facility	Supported
7	26	TS	O	73	File creation date/time	Supported
8	40	ST	O	74	File security	Not used
9	40	ST	O	75	File name/ID/type	Supported
10	80	ST	O	76	File comment	Not used
11	20	ST	O	77	File control ID	Not used
12	20	ST	O	78	Reference file	Not used

FTS Segment

SEQ	LEN	DT	R/O	ITEM#	ELEMENT NAME	ELR Usage
1	10	NM	O	79	File batch count	Supported
2	80	ST	O	80	File trailer	Not used

BHS Segment

SEQ	LEN	DT	R/O	ITEM#	ELEMENT NAME	ELR Usage
1	1	ST	R	67	Batch field separator	Supported "|"
2	5	ST	R	68	Batch encoding characters	Supported "^~\&"
3	20	ST	O	69	Batch sending application	Supported
4	20	ST	O	70	Batch sending facility	Supported
5	20	ST	O	71	Batch receiving application	Supported
6	20	ST	O	72	Batch receiving facility	Supported
7	26	TS	O	73	Batch creation date/time	Supported
8	40	ST	O	74	Batch security	Not Used
9	40	ST	O	75	Batch name/ID/type	Supported
10	80	ST	O	76	Batch comment	Not used
11	20	ST	O	77	Batch control ID	Not used
12	20	ST	O	78	Reference batch control ID	Not used

BTS Segment

SEQ	LEN	DT	R/O	RP#	ITEM#	ELEMENT NAME	ELR Usage
1	10	NM	O		93	Batch message count	Supported
2	80	ST	O		94	Batch comment	Not Used
3	100	NM	O	Y	95	Batch totals	Not Used

 ©Iowa Department of Public Health, 2021

MSH Segment

SEQ	LEN	DT	R/O	RP	TBL#	ITEM#	ELEMENT NAME	ELR Usage
1	1	ST	R			1	Field separator	
2	5	ST	R			2	Encoding characters	
3	225	HD	R			3	Sending application	Expecting lab system name
3.1	20	IS					Sending Application Name	
3.2	199	ST					Sending Application ID	
3.3	6	ID					Sending Application ID Type	
4	225	HD	R			4	Sending facility	|lab name^ CLIA code^CLIA| OR |Labname^OID^ISO|
4.1	20	IS					Sending Facility Name	
4.2	199	ST					Sending Facility ID	
4.3	6	ID					Sending Facility ID Type	
5	225	HD	R			5	Receiving application	
5.1	20	IS					Receiving Application Name	IA.DOH.IDSS
5.2	199	ST					Receiving Application ID	2.16.840.1.114222.4.3.3.19
5.3	6	ID					Receiving Application ID Type	ISO
6	225	HD	R			6	Receiving facility	
6.1	20	IS					Receiving Facility Name	IA DOH
6.2	199	ST					Receiving Facility ID	2.16.840.1.114222.4.1.3650
6.3	6	ID					Receiving Facility ID Type	ISO
7	26	TS	R			7	Date/Time of message	Required for ELR
8	40	ST	O			8	Security	Not supported
9	13	MSG	R		76, 3	9		
9.1	3	ID					Message Type	ORU
9.2	3	ID					Trigger Event	R01
9.3	7	ID					Message Structure	ORU_R01
10	199	ST	R			10	Message control ID	Construct the Message Control ID by using the sending application OID/CLIA number +'-+ date.timestamp down to the milliseconds
11	1	PT	R			11	Processing ID	Generally, `T' Test or `P' Production
12	5	VID	R		104	12	Version ID	2.3.1

Typical Message Sequence

Table 10 describes a typical segment ordering for a batch of messages. Note that the treatment of batches of XML encoded HL7 messages is somewhat more complicated. XML batches are described in the section entitled "Special Case of HL7 XML Encoding for Batches."

The file represented in Table 10 contains messages from two different hospitals that share an information system. Messages should never be interleaved when creating a batch file comprised of data from multiple hospital systems. To prevent this, each hospital's ORU messages must be batched together and bounded by Batch Header/Tail Segment pairs so that the messages' origins are clear. In the example, Institution A's messages are only found within the confines of the batch defined by the BHS on line 2 and the BTS on line 9.

1	FHS	Start Message File
2	BHS	Start Institution A's Messages
3	MSH	
4	ORU[1]	
5	MSH	
6	ORU	
7	MSH	
8	ORU	
9	BTS	End Institution A's Messages
10	BHS	Start Institution B's Messages
11	MSH	
12	ORU	
13	BTS	End Institution B's Messages
14	FTS	End Message File

■ Table 10 - Typical HL7 Batch Protocol Segment Usage

Special Case of HL7 XML Encoding for Batches

The HL7 2.3.1 XML specification defines the method for handling message batches in the **batch.dtd** file. The batch DTD introduces three new bounding tags that are used to structure the XML batch. The top-level tag is the BATCH tag, which wraps the entire contents of the XML batch file. The batch tag is synonymous with the FHS/FTS segment pair in that it only occurs once in the file and it bounds all batches in the file. Note that the BATCH tag does not replace the FHS and FTS tags- they still remain at the start and the end of the file.

The next level of organization is provided by the MESSAGEBATCH, which is used to delineate between the batches within the file. The MESSAGEBATCH is synonymous with the BHS and BTS segments. Every MESSAGEBATCH will contain one and only one BHS/BTS pair. One MESSAGEBATCH tag will be included for every institution's data being reported in the file.

The actual ORU messages are encapsulated within a standard XML CDATA block. The CDATA block is contained within a MESSAGES tag. There will be only one MESSAGES tag per MESSAGEBATCH. Figure 3 depicts the tags and structure required of an XML encoded HL7 batch file.

```
    <?xml version="1.0" encoding="UTF-8"?>
    <BATCH xmlns="urn:hl7-org:v2xml">
<FHS>
        <FHS.1>|</FHS.1>
        <FHS.2>^\&amp;</FHS.2>
        <FHS.3>HEALTHBRIDGE</FHS.3>
        <FHS.4>MATER MISERICORDIAE HOSPITAL</FHS.4>
        <FHS.6>Mouse, Mary</FHS.6>
        <FHS.7>
        <TS.1>20031126131700</TS.1>
        </FHS.7>
        <FHS.11>201</FHS.11>
</FHS>
<MESSAGEBATCH>
        <BHS>
```

[1] Note that in this example ORU should be taken to mean all the segments present in a valid ORU message.

```
                    <BHS.1>|</BHS.1>
                    <BHS.2>^\&amp;</BHS.2>
                    <BHS.7>
                    <TS.1>20031126131700</TS.1>
                    </BHS.7>
                    <BHS.11>77</BHS.11>
                </BHS>
                <MESSAGES><![CDATA[
                *Place multiple ORU messages here*
                ]]></MESSAGES>
                <BTS>
                    <BTS.1>3</BTS.1>
                </BTS>
            </MESSAGEBATCH>
            <FTS>
                <FTS.1>1</FTS.1>
            </FTS>
</BATCH>
```

- Figure 3 - Batch Protocol with HL7 XML Encoding

Identifying ID Types in PID3

The PID3 segment is reserved for one or more patient identifiers. Each PID3 segment submitted must have its PID3.2 field populated with an ID type identifier as specified by the HL7 vocabulary lists. Patient Identifier populated in this field should be generated **only once per patient** and is always unique to the patient. IDPH currently recognizes the ID type identifiers provided in Table 11.

ID Type	Identifier
Medical Record Number	MR
Patient Internal Identifier	PI

- Table 11 - PID3 ID Type Identifiers

Using OBX.2 to Identify OBX5's Data Type

The OBX.5 component of the OBX segment is defined as a "Varies," which is a data type similar to a variant. In order for the OBX segment to be processed correctly, the OBX.2 component must be used to record the data type so that the message can be processed by the receiving HL7 interface. OBX.2 may have one of the following values: SN, CWE, or DT.

Use of OBX.5 for Numeric Results

To simplify the structure of the IDPH Simplified ELR Message, the OBX segment was defined so that one SMF OBX segment instance could contain a local result code (an interpreted result) and a numeric result value (the data backing the interpretation). In this situation, a laboratory may use the "Abnormal flag" (Interpretation flag) field, OBX.8, to report the interpretation and use OBX.5 to report a quantitative result in a single OBX segment.

 ©Iowa Department of Public Health, 2021

Use of OBX.5 for Relevant Clinical Information

To simplify the structure of the IDPH Simplified ELR Message, some elements were flattened and relocated to either the PID segment or the OBR segment. These elements are clearly marked in the message definition tables by their OBX 5 HL7 Component designation. The original CDC specification indicates that each of these elements is recorded within its own OBX segment in component OBX.5. All relevant clinical information spawned OBX segments should be grouped under the corresponding OBR record.

For each case where you are creating a new OBX record to reflect an additional patient demographic you will need to incorporate LOINC codes to identify the exact demographic you are reporting. Table 12 lists a LOINC code for each of the additional demographic fields specified in the PID and OBR segments above.

Code	Description/Demographic	SMF Column Name
11368-8	ILLNESS/INJURY ONSET DATE/TIME	OBR – Onset Date
11449-6	PREGNANCY STATUS	PID – Pregnancy

- Table 12 - LOINC Codes used to identify Demographic Information carried in OBX segments

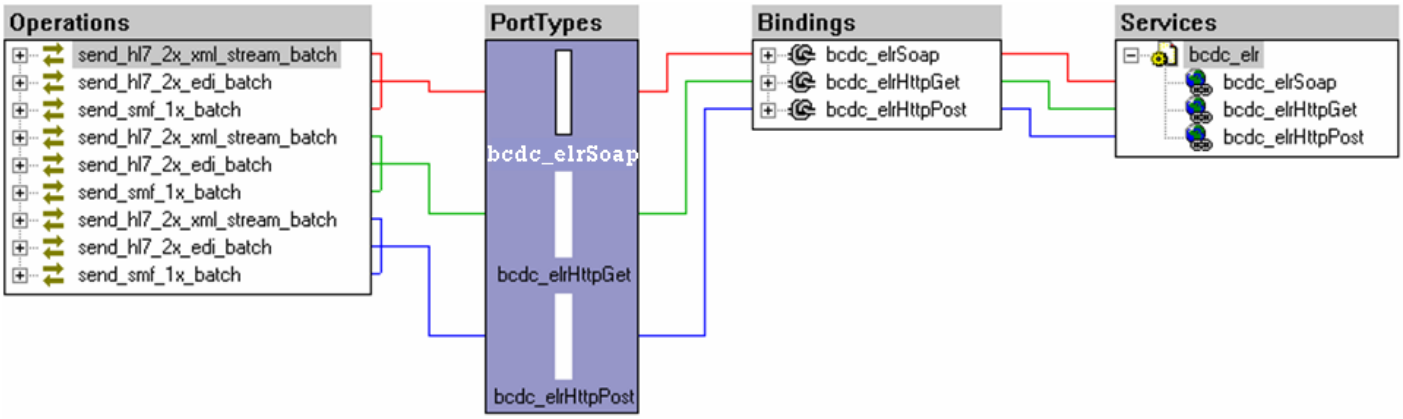
 ©Iowa Department of Public Health, 2021

Addendum: Web Service Protocols

Regardless of the messaging format used (SMF, HL7-EDI, HL7-XML) ELR messages will be transmitted to IDPH through the use of a web service. Web services provide the following benefits:

- Secure communications over HTTP/S (128-bit SSL)
- Client authentication and authorization
- Full automation of message transmission process
- Acknowledgement responses for communicating information about message processing results
- Support for all IDPH approved messaging formats

The web service published by IDPH uses the SOAP protocol with MIME bindings to allow for the transmission of file attachments. Transmitting a message batch to IDPH only requires an authentication step and a single web service method invocation. At the end of the method invocation the web service will return a result message indicating successful receipt of the file or an error code.



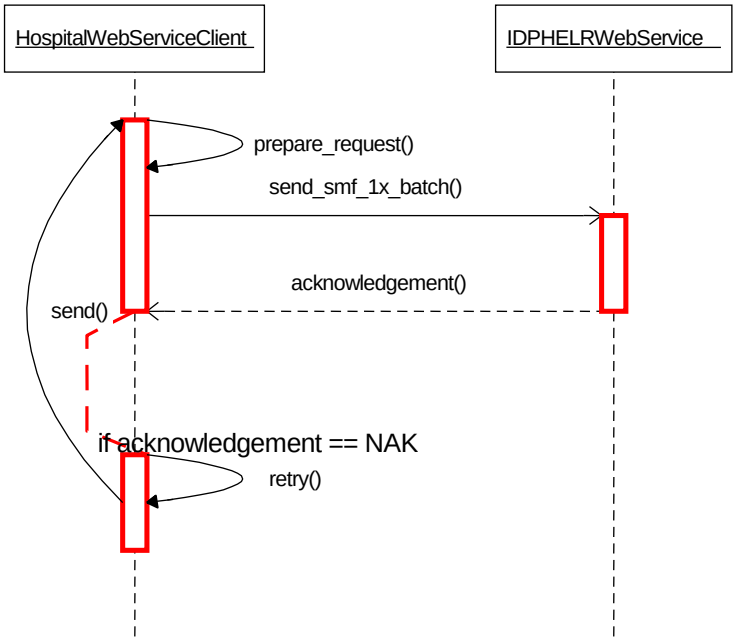
- Figure 4 - WSDL Schematic

Three basic operations are offered via SOAP, HTTP GET and HTTP POST style web services. Individual hospitals need only be concerned with the operation dedicated to the messaging format used by the hospital. As represented in Figure 4, the three operations (represented once for each web service style) are as follows:

Operation Name	Purpose
send_smf_1x_batch	Used to send batches of IDPH SMF 1.0 (or later) formatted messages.
send_hl7_2x_xml_stream_batch	Used to send batches of HL7 2.3.1 ORU messages using XML formatting.
send_hl7_2x_edi_batch	Used to send batches of HL7 2.3.1 ORU messages using EDI formatting.

- Table 13 - Web Service Operation Definitions

The implementations of the interaction between a hospital and the Web Service will vary based on the web service client technologies a hospital chooses. The interaction process is fundamentally a basic request-response pattern. Figure 5 contains a sequence diagram that describes the basic steps involved in a web service interaction between a hospital and the IDPH ELR web service.



- Figure 5 - Simple Web Services Interaction Sequence

Please be aware that the hospital web service client is required to retry sending a batch if a negative acknowledgement is received. Figure 5 represents the failure message as NAK but in reality, there are five possible error response codes and a retry is not appropriate in all cases. Some error messages are indicative of configuration problems, either locally or with the provider profile accessible on the ELR web site. Table 14 lists all possible web service response messages and indicates if an automatic retry is appropriate.

Message	Retry	Translation
SUCCESS	N	The submitted batch has been stored successfully
ERR_NOT_AUTHORIZED	N	The calling web service client is not authorized at this time to invoke the interface; the submitted batch is not stored
ERR_INVALID_PROTOCOL	N	The provider preferences do not allow for its client to invoke the web service; the submitted batch is not stored
ERR_INVALID_FORMAT	N	The acceptable file format set the in provider preferences does

		not match the called web service interface; the submitted batch is not stored
ERR_IO	Y	The web service encountered an error while storing the batch; the submitted batch is not stored
ERR_UNKNOWN	Y	The web service encountered an unexpected error; the submitted batch is not stored

- Table 14 - ELR System Response Messages

For those hospitals capable of developing a web services client, the first step is to obtain the Web Services Definition Language (WSDL) file that describes the ELR web services. The WSDL file can be found at:

https://elr.dph.state.ma.us/ws/d1/bcdc_elr.jws?WSDL

You need a valid IDPH provided web service login to access the URL.

Once you have a copy of the WSDL you can use it as a guide for your own custom web service development or as an input to a SOAP with Attachments (SwA) compliant web service stub generation program like Apache's wsdl2java (ws.apache.org/axis).

While the MIME attachment style is easiest to develop with Java based SOAP technologies, a pure HTTP Post solution to transmit a message can be implemented in any language. The following overview of an ELR transmission is provided to give insight to those implementing a true SOAP client and those using an HTTP style batch upload. The text provided in the example can be used as a template for creating a HTTP Post.

Table 15 provides an example depicting the data contained in the transmission of an ELR batch. The first column contains annotations useful for referring to specific message parts. The annotations would not be seen in an actual message. The message content is contained in the second column. The table itself is only used to organize the information; in an actual transmission, the data in the second column would be seen as one contiguous stream. The majority of the text in the second column is boilerplate. The **bold** text in the second column is text that would be changed by the reporting entity (more on this to follow).

The example logically divides the message up into three sections. The first section, represented by the first table row (lines 1-10) contains the HTTP parameters. These parameters are critical to the successful negotiation of the communication between the client and the web service. The only non-standard HTTP parameter is the last (line 10) and is required by the SOAP protocol to successfully route the message to the correct SOAP operation. Reporting entities must change the last portion of the SOAPAction URL to the operation name appropriate for the message batch encoding style being transmitted. A list of valid options for the operation name is provided in Table 13.

The *Content-Type* (line 2) parameter is critical to the correct handling of the message attachment. The first two elements *multipart/related; type="text/xml";* tell the server to expect a message that adheres to the multipart/related protocol and that the message payload will be either text or xml. Multipart/related messages require that each message part contain a Content-Id to uniquely identify the part within the message. The *Content-Type start* element makes reference to the *Content-Id* of the first message part. If the value found for *start* is not associated with any message parts, the message will fail to process. Finally, message parts are delimited using a simple string delimiter. The *boundary* defines the delimiter.

The second section in the table contains the SOAP message part. It starts with the boundary delimiter on line 11 and is followed by three parameters that describe the part. The third parameter

Content-Id (line14) is the critical unique identifier for the part. Note that the value for *Content-Id* on line 14 is identical to the value for *start* in the *Content-Type* parameter on line 2. This identifies the part as the first in the message.

HTTP	**1**	POST /ws/d1/bcdc_elr.jws HTTP/1.0
	2	Content-Type: multipart/related; type="text/xml"; start="<3D408EDC5A485D0AEE8CDB3A7105E29D>"; boundary="----=_Part_0_12710158.1136915548203"
	3	Accept: application/soap+xml, application/dime, multipart/related, text/*
	4	User-Agent: Axis/1.3
	5	Host: elr.dph.state.ma.us
	6	Cache-Control: no-cache
	7	Pragma: no-cache
	8	Content-Length: 2164
	9	Authorization: Basic **bHRiaS53ZWJzZFzc3dvcmQ=**
	10	SOAPAction: http://www.openuri.org/**send_smf_1x_batch**
SOAP	**11**	------=_Part_0_12710158.1136915548203
	12	Content-Type: text/xml; charset=UTF-8
	13	Content-Transfer-Encoding: binary
	14	Content-Id: <3D408EDC5A485D0AEE8CDB3A7105E29D>
	15	<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><soapenv:Body><**send_smf_1x_batch** xmlns="http://www.openuri.org/"><dh href="cid:A0D00719EDA95AB72853A1CA02E11E95"/></**send_smf_1x_batch**></soapenv:Body></soapenv:Envelope>
BATCH	**16**	------=_Part_0_12710158.1136915548203
	17	Content-Type: application/octet-stream
	18	Content-Transfer-Encoding: binary
	19	Content-Id: <A0D00719EDA95AB72853A1CA02E11E95>
	20	**BHS|1.2|State Laboratory Institute|22D0650270|20051020|2005102022D0650270** **PID||||||||JOE|DOE|||19701118000000|||1 WINTHROP ST||PLEASANT TOWN||00000||||||||** **ORC|INT MED HEALTH ASSESS|||||||(555)555-5555||||||** **OBR|10137|1648-5|20041012000000|||||||||SKN||||F|||||** **OBX|1648-5||25|mm|< .05||F|20041112000000|**
	21	------=_Part_0_12710158.1136915548203--

■ Table 15 - Example HTTP Post containing a SOAP message

The SOAP part's payload (line 15) is a complete XML document. Here again reporting entities will need to modify the operation name to correspond to the message batch encoding style. Note that the tag nested in the operation tag is a reference to the name of a parameter (*dh)* defined for the operation. Also notice that the parameter element contains no body and only one attribute. The attribute is an *href* with the value "*cid:A0D00719EDA95AB72853A1CA02E11E95*" which when read by the server is used as a pointer to the message part that should be provided as the value for the *send_smf_1x_*batch operation's *dh* parameter.

The last section (lines 16-21) starts with the boundary on line 16 and contains the batch file. Note that its *Content-Id* matches exactly with the value referenced in the *dh* element's *href*. The batch content starts on line 20 and must follow the formatting rules appropriate to the chosen encoding

style (SMF, HL7-EDI, or HL7-XML). The message must be terminated with a boundary (line 21) to signal its end.

The software that produced the message dynamically generated all the unique identifiers used in the example. If you decide to create your own HTTP style transmission software, you can choose any identifiers you want as long as the reference requirements mentioned above are satisfied. For instance, the first message part could be named *"soap_body"* and the second message part *"batch_file"* as long as the *start* attribute in the *Content-Type* parameter is set to "*soap_body"* and the *href* attribute of the *dh* element is set to *"batch_file."*

If you decide to implement an HTTP Post style client, remember that IDPH only accepts encrypted HTTP/s connections, so please factor that into your design and implementation.

 ©Iowa Department of Public Health, 2021